**Problem#2: Merchant Prediction**

Contents

[Data: 2](#_Toc489214287)

[Observations: 2](#_Toc489214288)

[Program & Detail: 3](#_Toc489214289)

[Solution: 3](#_Toc489214290)

[Data Preparation: 3](#_Toc489214291)

[Tune and Run the model: 3](#_Toc489214292)

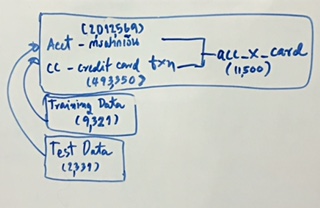
[Prediction and Output transformation: 4](#_Toc489214293)

[Improvement Plan: 4](#_Toc489214294)

[Reference: 4](#_Toc489214295)

# Data:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Type** | **File Name** | **Table name** | **Size (rows)** | **Filed Number** |
| Info Data | tj\_02\_account\_transaction.csv | acc | 2,012,569 | 6 |
|  | tj\_02\_credit\_card\_transaction.csv | cc | 493,350 | 6 |
|  | tj\_02\_acc\_x\_card.csv | acc\_x\_cc | 11,652 | 2 |
| training data | tj\_02\_training.csv | train.org | 9,321 | 2 |
| testing data | tj\_02\_test.csv | test.org | 2,331 | 1 |



# Observations:

* File name: tj\_02\_credit\_card\_transaction.csv
  + There are duplicated records in credit card transactions. It would be curious If it’s possible that consumers spend the same amount of money to the same merchant within an hour. However, I decided to keep these records. I think this kind of incident is able to be happened within an hour. It can be remove once I get more information such as txn\_min or txn\_sec.
* File name: tj\_02\_acc\_x\_card.csv
  + There are 4059 records are not associated with credit card number. So, they have been excluded from the table.
  + There are 226 credit card numbers which each card number is associate with 2 accounts. After merge table, these credit cards will have duplicated card data for associated accounts.

# Program & Detail:

|  |
| --- |
| platform x86\_64-w64-mingw32  arch x86\_64  os mingw32  system x86\_64, mingw32  status  major 3  minor 4.1  year 2017  month 06  day 30  svn rev 72865  language R  version.string R version 3.4.1 (2017-06-30)  nickname Single Candle |

# Solution:

## Data Preparation:

* acc table – change column name.
  + ‘from\_to\_account\_no’ to ‘from\_to’
  + ‘txn\_dt’ to ‘txn\_date’
* cc table – add column ‘txn\_type’ and insert “CC” data.
* acc\_x\_cc table – exclude all null credit card number records.
* Join cc table with acc\_x\_card.
* Concatenate cc table with acct table.
* Train table – add column name.
* Test table – add column name.
* Extract more 3 columns from txn\_type – is\_DR, is\_CR and is\_CC. This technique is help to increase more accuracy of the XGBoost model.
* Update all character data to be numeric. (XGBoost can accepted only numeric.)
* Update all NULL to be zero.
* Update all logical data (TRUE,FALSE) to be (1,0).
* Setup station of columns setting to be easy when tuning the model.
* Merge train and test data with all acct and cc data (matched by account\_no field).
* Convert data to xgboost format.

## Tune and Run the model:

* Set up Param as a list of parameters set up for xgboost.
* Set seed to ensure that the random number generated can be reproduced.
* Set up xgb.train such as data and nrounds.

## Prediction and Output transformation:

* Do prediction on the test account\_no. The output will be probability of is\_merchant. So, I have to convert this data to be a merchant flag.
* To convert probability to be merchant flag.
  + Setup cutoff number using median of all probability data.
  + Convert each record to be logical.
    - If probability > = cutoff number, then logical =1
    - If probability < cutoff number, then logical =0
  + Assign merchant flag for each account\_no using mode.
* Rearrange the merchant flag order to be matched with original test file before exporting.

# Improvement Plan:

* According to train data observation, it seems that the merchant accounts have high number of DR transactions. So, I’ve planned to count number of flag TRUE in is\_DR, is\_CR and is\_CC columns. Group them by account\_no, txn\_amount,txn\_month and txn\_day. Create new model using this new data set and observe accuracy after merging the prediction outcome of the existing model with this model.
* Do the same approach as above plan but change the group by to account\_no, txn\_month.

# Reference:

https://www.analyticsvidhya.com/blog/2016/01/xgboost-algorithm-easy-steps/